Advances in Surface Meshing (Computational Geometry and Discrete Mathematics)

AUTHOR(S):

Cheng, Siu-Wing

CITATION:

Cheng, Siu-Wing. Advances in Surface Meshing (Computational Geometry and Discrete Mathematics). 数理解析研究所講究録 2009, 1641: 125-134

ISSUE DATE:

2009-04

URL:

http://hdl.handle.net/2433/140578

数理解析研究所講究録
第 1641 巻 2009 年 125-134

Advances in Surface Meshing

Siu-Wing Cheng*

Abstract

Meshing a surface means approximating it by a polygonal surface with triangular faces. We outline the recent progress on providing theoretical guarantees and meshing more complex types of surfaces. We also discuss some future research problems.

1 Introduction

Meshing a surface means approximating it with a polygonal surface with triangular faces. It is a ubiquitous task in geometric applications such as rendering, numerical simulation, and computer aided design. It is a challenging problem because of the numerical, topological, and geometrical issues involved. First, one has to sample points on the input surface which serve as the mesh vertices. Second, one has to connect the mesh vertices to produce the correct topology and desired geometry. The first step of sampling points inevitably requires solving some systems of equations (involving the equation of the input surface). Solving systems of equations is by itself a big research topic and so we will assume this capability for a system of equations of constant size throughout this article. Still, one has to formulate the right systems of equations to be solved. Our focus is on the second issue of producing the correct topology and desired geometry. We concentrate on algorithms following the Delaunay refinement paradigm, which has been recently shown to work for piecewise smooth complexes as well. If a smooth closed surface is to be meshed, alternatives are known including cube-based algorithms [10, 13, 12], a sweep algorithm for algebraic surfaces [11], and algorithms using Morse Theory [1, 14].

We first introduce two Delaunay refinement algorithms for meshing a smooth closed surface, one by Boissonnat and Oudot [2] and one by Cheng et al. [6]. Then, we sketch the extension by Cheng, Dey, and Ramos [5] to mesh a piecewise smooth complex. If necessary, Delaunay refinement can be repeated for the algorithms above to obtain triangles of bounded aspect ratio. We conclude with some discussions and open problems.

2 Smooth closed surface

Let $\Sigma \subset \mathbb{R}^3$ be a smooth compact surface without boundary. Without loss of generality, assume that Σ is connected. Let $n(x)$ denote the outward unit normal of Σ at a point

*Department of Computer Science and Engineering, HKUST, Clear Water Bay, Hong Kong, scheng@cse.ust.hk

$x \in \Sigma$. Given a set of points S on Σ, let $\operatorname{Del} S$ denote the Delaunay triangulation of S and let $\operatorname{Del} S|_\Sigma$ denote the restricted Delaunay triangulation which consists of Delaunay edges and triangles whose dual Voronoi faces intersect Σ. For any point $p \in S$, let $\operatorname{star}(p, \Sigma)$ denote the set of edges and triangles in $\operatorname{Del} S|_\Sigma$ incident to p. A ball B is a surface Delaunay ball if B is empty, the center of B lies on Σ, and the boundary of B passes through three points in S. Clearly, the center of a surface Delaunay ball belongs to some Voronoi edge. The medial axis is the set of centers of all maximal balls whose interiors avoid Σ. The local feature size of a point $x \in \Sigma$, $\operatorname{lfs}(x)$, is the distance between x and the medial axis of Σ.

2.1 Meshing via a lfs lower bound

The set S is a loose ε-sample of Σ if every surface Delaunay ball has radius at most $\varepsilon \operatorname{lfs}(x)$, where x is the ball center. Boissonnat and Oudot proved the following result [2].

Lemma 2.1 *Assume that S is a loose ε-sample for some $\varepsilon < 0.1$ and that $\operatorname{Del} S|_\Sigma$ contains at least one triangle. Then $\operatorname{Del} S|_\Sigma$ is isotopic to Σ.*

This motivates the following strategy. Let ε be some constant less than 0.1. It is assumed that the initial S contains of three given points a, b and c such that: (i) $\operatorname{Del} S|_\Sigma$ contains abc; (ii) the radius of the smallest surface Delaunay ball of abc is at most $\varepsilon \operatorname{lfs}(x)/3$, where x is the ball center; (iii) a function $\varphi : \Sigma \to \mathrm{R}$ is given such that $0 < \varphi(x) \leq \operatorname{lfs}(x)$. Notice that condition (ii) does not exclude the possibility that abc may have a surface Delaunay ball of radius larger than $\varepsilon \operatorname{lfs}(x)$. Indeed, Boissonnat and Oudot's algorithm repeatedly inserts large surface Delaunay ball centers as follows.

1. Find a surface Delaunay ball B with radius greater than $\varepsilon \varphi(x)$.
2. If B exists, insert the center of B into S, update $\operatorname{Del} S|_\Sigma$, and go back to step 1. Otherwise, output $\operatorname{Del} S|_\Sigma$.

Finding surface Delaunay ball centers is equivalent to computing the intersections between the Voronoi edges and Σ. Because abc is so small, it can be shown that abc persists in $\operatorname{Del} S|_\Sigma$ throughout the algorithm. Thus, if the algorithm terminates, Lemma 2.1 ensures that $\operatorname{Del} S|_\Sigma$ is topologically correct. Termination is guaranteed because Σ is compact and the distance between a newly inserted point x and any existing point in S is more than $\varepsilon \varphi(x)$.

2.2 Meshing via the topological ball property

The Voronoi diagram of S, $\operatorname{Vor} S$, has the topological ball property if every k-dimensional Voronoi face either avoids Σ or intersects Σ generically in a $(k-1)$-dimensional topological ball. That is, if a Voronoi edge or facet or cell intersects Σ, the intersection is a single point or arc or topological disk, respectively. The following result by Edelsbrunner and Shah [8] explains why the topological ball property is useful.

Lemma 2.2 *If $\operatorname{Vor} S$ has the topological ball property, $\operatorname{Del} S|_\Sigma$ is homeomorphic to Σ.*

While Boissonnat and Oudot's algorithm is based on achieving certain sampling density, the algorithm of Cheng et al. [6] is based on Lemma 2.2. We present a slightly different version here to facilitate the extension to piecewise smooth complexes later. Let lfs^* be the minimum local feature size over Σ. Roughly speaking, violations of the topological ball property are detected and in case of violation, there exists a point x in $\text{Vor}\, S \cap \Sigma$ at distance more than $\omega\, \text{lfs}^*$ from the existing points in S for some constant ω. The algorithm repeatedly identifies and inserts such a point until the topological ball property is satisfied. Termination follows again from the compactness of Σ and the interpoint distance lower bound.

Primitives. The algorithm uses the following numerical primitives. For each Delaunay simplex σ, we use V_σ to denote the dual Voronoi face.

- $\text{MulInt}(p \in S,\, \Sigma)$: If $|V_t \cap \Sigma| = 1$ for any triangle $t \in \text{star}(p, \Sigma)$, $\text{MulInt}(p)$ returns null; otherwise, $\text{MulInt}(p)$ returns the furthest intersection point from p between the edges of V_p and Σ.
- $\text{SurNorm}(\omega \leq 0.01,\, p \in S,\, \Sigma)$: Let $\phi \in (4\omega, \pi/16)$ be a constant. If $\angle n(p), n(z) < \phi$ for any point $z \in V_p \cap \Sigma$, $\text{SurNorm}(p)$ returns null; otherwise, it returns a point $z \in V_p \cap \Sigma$ such that $\angle n(p), n(z) = \phi$.
- $\text{CurNorm}(\omega \leq 0.01,\, p \in S,\, \Sigma)$: Let ψ be a constant chosen from $(4\omega + \phi, 4\omega + \pi/16)$. $\text{CurNorm}(p)$ returns a point $y \in V_e \cap \Sigma$ for some edge $e \in \text{star}(p, \Sigma)$ such that $\angle \vec{d_1}, \vec{d_2} = \psi$, where $\vec{d_1}$ and $\vec{d_2}$ are the projections of $n(p)$ and $n(y)$ onto the plane of V_e, respectively. $\text{CurNorm}(p)$ returns null if such a point does not exist.
- $\text{NoDisk}(p \in S,\, \Sigma)$: If $\text{star}(p, \Sigma)$ does not form a topological disk, $\text{NoDisk}(p)$ returns the furthest intersection point between the edges of V_p and Σ. Otherwise, it returns null.

Implementation of primitives. We give the implementations of the primitives, which involve formulating and solving the appropriate systems of equations. Let $E(x) = 0$ denote the equation of Σ.

- $\text{MulInt}(p \in S,\, \Sigma)$: For any triangle $t \in \text{star}(p, \Sigma)$, V_t is in the intersection of two planes and let $H_{t,1} = 0$ and $H_{t,2} = 0$ denote the equations of them. We solve the system: $E(x) = 0$, $H_{t,1} = 0$, $H_{t,2} = 0$ and we select those solutions that lie on V_t. These are the intersection points between V_t and Σ. If there are more than one, we return the one furthest from p.
- $\text{SurNorm}(\omega \leq 0.01,\, p \in S,\, \Sigma)$: Define $G(x) = \langle \nabla E(p), \nabla E(x) \rangle - \|\nabla E(p)\| \cdot \|\nabla E(x)\| \cdot \cos\phi$. Ignoring the degenerate cases, the system: $E(x) = 0$, $G(x) = 0$ describes a collection $\mathcal{F}$ of disjoint smooth closed curves. We solve for the set of intersections between $\mathcal{F}$ and the support planes of V_p. If any intersection obtained belongs to V_p, return it. If no intersection in V_p is obtained, we still need to check the closed curves in completely inside V_p. We solve the system: $E(x) = 0$, $G(x) = 0$,

$\langle \nabla G(x) \times \nabla E(x), (p-x) \rangle = 0$. The solutions are the tangential contact points with $\mathcal{F}$ and balls centered at p. If any tangential contact point belongs to V_p, return it. If no contact point is found in V_p, $\mathcal{F}$ avoids V_p and we return null.

- CurNorm($\omega \leq 0.01$, $p \in S$, Σ): For each edge $e \in \text{star}(p, \Sigma)$, let q_e denote a point on the support plane of V_e; let n_e denote a unit vector orthogonal to the support plane of V_e; and define the function $G_e(x) = \nabla E(x) - \langle \nabla E(x), n_e \rangle \cdot n_e$. Notice that $G_e(x)$ is the projection of the vector $\nabla E(x)$ onto a plane orthogonal to n_e. We solve the system: $E(x) = 0$, $\langle (x - q_e), n_e \rangle = 0$, $\langle G_e(x), G_e(p) \rangle = \|G_e(x)\| \cdot \|G_e(p)\| \cdot \cos \psi$. If any solution obtained belongs to V_e, return it. Return null if no solution can be returned for any edge in $\text{star}(p, \Sigma)$.

- NoDisk($p \in S$, Σ): NoDisk(p, σ) returns null if the union of triangles in $\text{star}(p, \Sigma)$ is a topological disk. Otherwise, we compute the intersection points between the edges of V_p and Σ and return the furthest one from p.

Algorithm. We first initialize S to contain a point on Σ. A convenient choice is a height critical point at which the outward normal is vertical. Afterwards, for every point $p \in S$, we call MulInt(p, Σ), SurNorm(ω, p, Σ), CurNorm(ω, p, Σ), and NoDisk(p, Σ) in this order. If a point x is returned in some call, then we stop, insert x into S, and update Del S and Vor S. Afterwards, we repeat the calling sequence for every point in S again until S does not grow anymore.

Topology. Each point returned by any of the primitives is more than $\omega\,\text{lfs}^*$ away from the existing points in S. The readers are referred to full proofs in [5, 6]. We explain why the topological ball property is achieved when the algorithm terminates.

Clearly, MulInt ensures that every Voronoi edge intersects Σ in a single point, if at all, upon termination.

Consider a Voronoi facet V_e. If $V_e \cap \Sigma$ is non-empty and not an arc, it may contain some closed curve and/or more than one arc. If $V_e \cap \Sigma$ contains a closed curve, CurNorm will be triggered to return a point. In case of two or more arcs, because MulInt does not return a point, these arcs have endpoints on four distinct edges of V_e. It follows that there are two triangles in $\text{Del}\, S|_\Sigma$ incident to e. This is a non-topological-disk feature which triggers NoDisk to return a point. Hence, every Voronoi facet intersects Σ in an arc, if at all, upon termination.

Consider a Voronoi cell V_p. Upon termination, $\text{star}(p, \Sigma)$ is a topological disk. Since MulInt and CurNorm return null, in the boundary of $V_p \cap \Sigma$ no two curves intersect the same edges of V_p and no curve lies within a facet of V_p. Thus, the boundary of $V_p \cap \Sigma$ must be a closed curve as it is dual to $\text{star}(p, \Sigma)$. Because SurNorm(p) returns null, $\angle n(p), n(z) < \pi/2$ for any point $z \in V_p \cap \Sigma$. It can be shown [6] that the projection of $V_p \cap \Sigma$ to a plane orthogonal to $n(p)$ is an injective map. The projected image is a planar bounded region with a closed boundary curve. So the image must be a topological disk. The injectivity makes the projection a homeomorphism. Hence, $V_p \cap \Sigma$ is a topological disk.

3 Piecewise smooth complex

The algorithm based on topological ball property can be extended to mesh piecewise smooth complexes (PSCs). A domain $\mathcal{D}$ is a PSC if it consists of smooth k-manifolds for $0 \leq k \leq 2$. We require each manifold to be C^3-smooth. The domain $\mathcal{D}$ satisfies the proper intersection requirement: (i) interiors of its element are disjoint; (ii) for any $\sigma \in \mathcal{D}$, the boundary of σ is a union of elements in $\mathcal{D}$; (iii) for any $\sigma, \tau \in \mathcal{D}$, $\sigma \cap \tau$ is empty or a union of elements in $\mathcal{D}$.

For each curve or surface $\sigma \in \mathcal{D}$, we assume that σ is part of a smooth closed curve or surface, respectively, denoted by $\mathrm{mani}(\sigma)$. If σ is a surface, the implicit equation of $\mathrm{mani}(\sigma)$ is given; if σ is a curve, $\mathrm{mani}(\sigma)$ is specified as the intersection of two smooth closed surfaces whose implicit equations are given. For each point x on an element $\sigma \in \mathcal{D}$, if σ is a surface, $n_\sigma(x)$ denotes the unit outward normal of $\mathrm{mani}(\sigma)$ at x; if σ is a curve, $n_\sigma(x)$ denotes the plane orthogonal to $\mathrm{mani}(\sigma)$ at x.

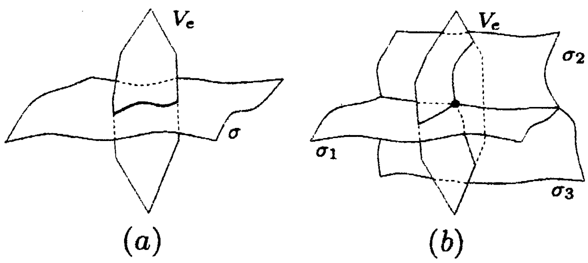

Figure 1: In (a), V_e intersects a 2-face in an arc. In (b), V_e intersects the curve in a single point and for $1 \leq i \leq 3$, $V_e \cap \sigma_i$ are arcs incident to this single point.

Given a set S of point samples from $\mathcal{D}$, there is an extended topological ball property [8] for $\mathrm{Vor}\, S$ which requires:

(i) Every Voronoi edge intersects $\mathcal{D}$ in at most a single point.

(ii) Let V_e be a Voronoi facet that intersects $\mathcal{D}$. The facet V_e either avoids a surface in $\mathcal{D}$ or intersects it generically in an arc. Also, V_e either avoids the curves in $\mathcal{D}$ or intersects exactly one of them, say σ. In the latter case, $V_e \cap \sigma$ is a single point and V_e intersects the surfaces incident to σ in arcs that meet only at $V_e \cap \sigma$. Figure 1 illustrates the cases.

(iii) For any point $p \in S$ and for any curve or surface $\sigma \in \mathcal{D}$, V_p intersects σ iff $p \in \sigma$. In this case $V_p \cap \sigma$ is an open curve or a topological disk, respectively.

Lemma 3.1 ([8]) *If* $\mathrm{Vor}\, S$ *has the extended topological ball property,* $\mathrm{Del}\, S|_{\mathcal{D}}$ *is homeomorphic to* $\mathcal{D}$.

Comparing Lemma 2.2 and Lemma 3.1, we would expect a similar meshing strategy should work for PSC. This is true, roughly speaking, for the interior of each surface in $\mathcal{D}$. The new challenge is to approximate the curves in $\mathcal{D}$ by polygonal curves at which the incident surface meshes must meet seamlessly. Our solution is to construct these polygonal curves first before meshing the surfaces. Afterwards, the vertices of the polygonal curves form the initial vertex set S and then we mesh the surfaces by fixing violations of the extended topological ball property. To make the surface meshes meet seamlessly, we need to ensure that every polygonal curve edge eventually appears as an edge in every incident surface mesh.

3.1 Curve meshing

Let σ be a curve in $\mathcal{D}$. Let u and v be the endpoints of σ. We initialize two protecting balls centered at u and v. Then, we march along σ from u and v, placing more balls on the way. These balls enjoy the following properties: (i) the ball radii vary slowly along σ; (ii) consecutive balls overlap; (iii) non-consecutive balls are separated by a gap; (iv) the curve tangent within a ball varies little; (v) the normals to surfaces incident to σ vary little inside every ball; (vi) every ball intersects mani(σ) in an arc and mani(τ) in a topological disk for every surface τ incident to σ.

Let $\omega \leq 0.01$ be a constant. Consider any point $x \in \sigma$. Let $B(x,r)$ denote a ball centered at x with radius r. Let $d_\omega(x)$ be the largest distance such that $\angle n_\sigma(x), n_\sigma(y) \leq \omega$ for any point $y \in$ mani(σ). Let $g(x)$ be the minimum distance from x to an element of $\mathcal{D}$ that does not contain x. Let $b(x)$ be the largest value such that $B(x, b(x)) \cap$ mani(σ) is an arc and $B(x, b(x)) \cap$ mani(τ) is a topological disk for any surface τ containing x. Then, we define

$$\mathrm{f}_\omega(x) = \min\{d_\omega(x), g(x), b(x)\}.$$

The value $\mathrm{f}_\omega(x)$ will be used to control the ball radius and its definition is clearly geared towards properties (iv)–(vi) above. However, $\mathrm{f}_\omega(x)$ may change very abruptly. Therefore, we have to do an on-the-fly Lipschitization along σ in order to achieve property (i).

Let $\lambda = 0.01$. We place two protecting balls $B_u = B(u, \lambda \mathrm{f}_\omega(u))$ and $B_v = B(v, \lambda \mathrm{f}_\omega(v))$ at u and v. Let $x_0 = \sigma \cap \mathrm{bd} B_v$, $x_1 = u$ and $x_2 = \sigma \cap \mathrm{bd} B_u$. Let $r_i = \lambda \mathrm{f}_\omega(x_i)$ for $i \in \{0, 1, 2\}$. The protecting ball B_{x_2} at x_2 is $B(x_2, r_2)$. For $k \geq 3$, we compute the intersection point x_k between the boundary of $B(x_{k-1}, 6r_{k-1}/5)$ and the portion of σ from x_{k-1} to v. Define

$$r_k = \max \begin{cases} \frac{1}{2}\|x_{k-1} - x_k\| \\ \min_{0 \leq j \leq k}\{\lambda \mathrm{f}_\omega(x_j) + \lambda\|x_j - x_k\|\} \end{cases}$$

and

$$r_{0k} = \min_{0 \leq j \leq k} \{\, r_j + \lambda\|x_j - x_0\| \,\}.$$

If $B(x_k, r_k) \cap B(x_0, r_{0k}) = \emptyset$, the protecting ball at x_k is

$$B_{x_k} = B(x_k, r_k).$$

Figure 2 shows an example of the construction of B_{x_k}. We force $r_k \geq \frac{1}{2}\|x_{k-1} - x_k\|$ so that B_{x_k} overlaps significantly with $B_{x_{k-1}}$.

We continue to march toward x_0 and construct protecting balls until the candidate ball $B(x_m, r_m)$ that we want to put down overlaps with $B(x_0, r_{0m})$. In this case, we reject x_m and $B(x_m, r_m)$. We set the protecting ball at x_0 to be $B_{x_0} = B(x_0, \frac{4}{5} r_{0\,m-1})$.

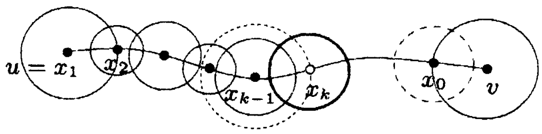

Figure 2: $B(x_{k-1}, 6r_{k-1}/5)$ and $B(x_0, r_{0k})$ are drawn with dashed circles. The bold circle denotes B_{x_k}.

To cover the gap between $B_{x_{m-1}}$ and B_{x_0}, a convenient choice is a ball orthogonal to $B_{x_{m-1}}$ and B_{x_0}. Compute the bisector plane of $B_{x_{m-1}}$ and B_{x_0} with respect to the power distance and the intersection point y_m between this plane and the portion of σ between x_{m-1} and x_0. The protecting ball B_{y_m} is the ball centered at y_m with radius $(\|x_0 - y_m\|^2 - \text{radius}(B_{x_0})^2)^{1/2} = (\|x_{m-1} - y_m\|^2 - \text{radius}(B_{x_{m-1}})^2)^{1/2}$.

Finally, we turn each protecting ball B_p into a weighted point p with weight $w_p = \text{radius}(B_p)^2$. Since adjacent protecting balls overlap, this gives the advantage that every edge connecting two adjacent weighted points is always weighted Delaunay. So these edges will persist throughout the repeated insertions of unweighted points in the rest of the algorithm.

3.2 Algorithm

Since we turn every protecting ball into a weighted point, it is natural to maintain a weighted Delaunay triangulation and a restricted weighted Delaunay triangulation. The balls on the curves in $\mathcal{D}$ are the only weighted points. All other points inserted subsequent to produce the mesh are unweighted. Lemma 3.1 works for the weighted Voronoi diagram and Delaunay triangulation as well. So we abuse the notation and use $\mathrm{Del}\, S$ and $\mathrm{Del}\, S|_\Sigma$ to denote the weighted triangulations.

We use the primitives in Section 2.2 plus one more primitive Infringe($p \in S$, surface $\sigma \in \mathcal{D}$). This primitive ensures that: (i) p is not connected to a vertex $q \notin \sigma$; (ii) if $p \in \mathrm{bd}\sigma$, p is connected to adjacent vertices in $\mathrm{bd}\sigma$ only. In the presence of such a violating edge pq, Infringe(p, σ) returns a point in $V_{pq} \cap \sigma$; otherwise, it returns null.

The whole algorithm works as follows. First, we mesh the curves as discussed in the previous subsection. Second, we insert a point in each surface in $\mathcal{D}$ outside the protecting balls on the curves. The points inserted so far form the initial vertex set S. Third, for every surface σ in $\mathcal{D}$ and for every point $p \in S$ on σ, we call Infringe(p, σ), MulInt(p, σ), SurNorm(ω, p, σ), CurNorm(ω, p, σ), and NoDisk(p, σ) in this order. If a point is returned in some call, then we stop, insert x into S, and update $\mathrm{Del}\, S$ and $\mathrm{Vor}\, S$. Afterwards, we repeat the third step until S does not grow anymore. We output $\mathrm{Del}\, S|_\mathcal{D}$ upon termination.

4 Other mesh qualities

We have concerned ourselves with topological correctness so far. In many applications it is important that triangles with small aspect ratio and their normals approximate the surface normals.

Aspect ratio. A small aspect ratio is equivalent to a small constant upper bound on the ratio of the circumradius to the shortest edge length. This is the so-called radius-edge ratio.

The two algorithms described for smooth closed surface can be enhanced by the same refinement strategy to give good triangle shape. For any restricted Delaunay triangle t, if the radius-edge ratio of t exceeds 1, we simply insert the insertion point between V_t and the surface. Notice that the distance between this newly inserted point and any existing point is no smaller than the shortest edge length of t. Thus, the interpoint distance lower bound is maintained. Of course, the topological correctness may have been disturbed by this insertion. So we need to rerun the meshing algorithm again. The whole process repeats until every triangle has radius-edge ratio less than 1. The smallest angle is thus at least $\pi/6$.

The same enhancement also works for the algorithm for PSC, except that we can only split triangles with unweighted vertices. The consequence is that no guarantee can be offered for the shape of triangles incident to surface boundaries.

Triangle normal. Since Boissonnat and Oudot's algorithm forces all surface Delaunay ball to be small with respect to local feature size, it is known from standard surface sampling theory that the triangle normals approximate the surface normals. The two algorithms based on the (extended) topological ball property can be enhanced further. Let p be a vertex on a surface σ and let t be a triangle in $\mathrm{star}(p, \sigma)$. If $\angle n_\sigma(p), V_t \geq 26\omega$, we insert the intersection point $V_t \cap \sigma$. As before, we need to rerun the meshing algorithm until there is no violation in the topology, triangle shape, or triangle normal.

With these mesh quality improvements, the PSC meshing algorithm has the following performance.

Theorem 4.1 *Let $\omega \leq 0.01$. Given a PSC $\mathcal{D}$, one can construct a weighted point set S such that:*

(i) $\mathrm{Del}\, S|_{\mathcal{D}}$ *is homeomorphic to* $\mathcal{D}$.

(ii) *Every output triangle with unweighted vertices has radius-edge ratio less than 1.*

(iii) *For any vertex p on a surface $\sigma \in \mathcal{D}$ and for any triangle $t \in \mathrm{star}(p, \sigma)$, $n_\sigma(p)$ makes an angle less than 26ω with the normal of t.*

(iv) *For any edge e in $\mathrm{Del}\, S|_{\mathcal{D}}$ with an unweighted endpoint, the two triangles incident to e make a dihedral angle greater than $\pi - 52\omega$.*

5 Discussion

Several problems awaits further research. The two meshing algorithms based on the topological ball property use expensive numerical primitives. The main question is whether one can obtain a result with less expensive numerical primitives. There is a solution by Cheng et al. [4] in this direction. The same curve meshing phase is kept, while the primitives SurNorm and CurNorm are eliminated. Instead, only a given upper bound r on the triangle size and a more sophisticated topological disk condition than NoDisk are used. No matter what r is, the output is always a valid complex formed by surface meshes. When r is sufficiently small, the output is homeomorphic to the input PSC. More recently, Dey and Levine [7] announces that the curve meshing can also be done adaptively without expensive numerical primitives. Still, it would be interesting to see if parametric surfaces can be handled (without resorting to implicitizing them first) because of their popularity in computer aided design.

A stronger topological guarantee for PSC meshing is isotopy. Does it already follow from the extended topological property in our case? Or is a vastly different approach necessary?

Another direction is to maintain a valid and good mesh when the points move with a deforming surface. There is a recent result in [3] saying that the update time can be made worst-case optimal (with respect to some worst-case surface and motion) in the absence of topological changes. Can topological changes be accommodated? Can something more be said about the update time with respect to the specific surface and motion at hand?

References

[1] J.-D. Boissonnat, D. Cohen-Steiner, and G. Vegter. Isotopic implicit surface meshing. *Proc. 36th Annu. ACM Sympos. Theory of Computing*, 2004, 301–309.

[2] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67 (2005), 405–451.

[3] S.-W. Cheng and T. K. Dey. Maintaining deforming surface meshes. *Proc. 19th ACM-SIAM Sympos. Discrete Alg.*, 2008, 112–121.

[4] S.-W. Cheng, T.K.Dey, and J.A. Levine. A practical Delaunay meshing algorithm for a large class of domains. *Proc. 16th Internat. Meshing Roundtable*, 2007, 477–494.

[5] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. To appear in *Discr. Comp. Geom.* Preliminary version in *Proc. 18th ACM-SIAM Sympos. Discrete Alg.*, 2007, 1096–1105.

[6] S.-W. Cheng, T. K. Dey, E. A. Ramos, and T. Ray. Sampling and meshing a surface with guaranteed topology and geometry. *SIAM J. Computing*, 37 (2007), 1199–1227.

[7] T.K. Dey and J.A. Levine. Delaunay meshing of piecewise smooth complexes without expensive predicates. Tech Report, OSU-CISRC-7108-TR40, July 2008.

[8] H. Edelsbrunner and N. R. Shah. Triangulating topological spaces. *Internat. J. Comp. Geom. Appl.*, 7 (1997), 365–378.

[9] F. Labelle and J.R. Shewchuk, Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graphics* 26(3), 2007.

[10] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Proc. SIGGRAPH*, 1987, 163–169.

[11] B. Mourrain and J.-P. Tècourt. Isotopic meshing of a real algebraic surface. Technical Report RR-5508, INRIA-Sophia Antipolis, France, 2005.

[12] S. Plantinga and G. Vegter. Isotopic approximation of implicit curves and surfaces. *Proc. 2004 Eurographics/ACM SIGGRAPH Sympos. Geom. Process.*, 2004, 245–254.

[13] J.M. Snyder. *Generative modeling for computer graphics and CAD: symbolic shape design using interval analysis.* Academic Press, 1992.

[14] B. T. Stander and J. C. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. *Computer Graphics*, 31 (1997), 279–286.